ADAP[illegible] CLOCK SYNCHRONIZATION SCHEMES FO[illegible]EAL-TIME TRAFFIC IN BROADBAND PACKET NETWORKS

R. P. Singh and S. H. Lee
Bell Communications Research, Inc., USA

ABSTRACT

A simple and inexpensive way of achieving synchronization in a broadband packet network is to use independent clocks at each node in the network with the same mean frequency, and adjust the destination clock frequency on the basis of monitoring the destination buffer level, which is a linear function of the frequency difference between the source and destination clocks. However, due to the inherently stochastic nature of packet transport, a periodic packet stream of real-time traffic - such as voice, video or a circuit-switched signal - tend to become quasi-periodic, causing random fluctuations in the destination buffer level. This network-induced jitter makes it difficult to extract source clock timing information from the arriving packet stream.

In this paper we illustrate the phenomenon of packet jitter via a simple example, and propose methods by which the destination clock frequency can be adjusted, so that asymptotically it follows the source clock frequency, while minimizing the effects of packet jitter on the outgoing packet stream. We provide a comprehensive asymptotic analysis of the estimation and control algorithms, and also discuss the implementation aspects of these algorithms.

1. INTRODUCTION

With the deployment of optical fiber and high speed, distributed switching into the public network, broadband packet techniques are emerging to be the most viable alternative for transporting real- and nonreal-time signals from diverse traffic sources, which range from low speed data to high speed motion pictures. In a broadband packet network, information bits from all user services as well as from the existing digital circuit-switched traffic are packed into slotted packets, which are transported in a truly integrated manner at a network backbone rate, e.g., 150 Megabits per second (Mbps). Asynchronous Time-Division (ATD) Network [1] and Dynamic Time-Division Multiplexed (DTDM) Network [2] are two prominent examples of a broadband packet network.

Real-time services generating a periodic packet stream may require stringent time transparency, which may call for synchronization of the broadband packet network. Providing synchronization through a common master clock is not a cost-effective option, although it may make the transport of a periodic packet stream much easier. Alternatively, the arriving periodic packet stream at the destination node may be used to provide the source clock timing information, and the destination clock may be adjusted to follow the source clock frequency by monitoring the packet filling level function at the destination buffer, so as to reconstruct the original traffic without any loss of information due to the frequency difference between the source and the destination clocks. Unfortunately, the arriving packet stream at the destination buffer is no longer periodic due to random delays introduced by the network on each packet of the periodic packet stream during transit. The aperiodicity of the packet arrival at the destination buffer appears as a random fluctuation in the destination buffer level, which cannot be directly used as a measure of the frequency difference between the source and destination clocks. Furthermore, this fluctuation may be correlated from packet to packet as well as nonstationary; its statistics may vary whenever there would be a change in the multiplexing and switching pattern due to addition or deletion of calls.

The central questions in the design of synchronization and clock recovery schemes are: (i) how to separate the feeble signal (the small frequency difference between the source and the destination clocks) from the noise (the packet jitter) with a large magnitude to obtain its estimate; and (ii) how to control the destination clock frequency using the frequency difference estimate. (The terms packet jitter and noise are used interchangeably in this paper.) The problem of synchronization and clock recovery in broadband packet networks has been mentioned in [3] in rather general terms, with no specific design details. The reference [4] is the first to have attempted to address these issues in a quantitative manner. But their assumptions are quite restrictive. Moreover, the deadbeat control law used to adjust the frequency difference in one time step will accentuate the residual estimation error, possibly leading to instability.

Our approach in this paper is based on the following main assumptions: the source clock is generally stable with an essentially constant frequency; there is no packet loss during the transit, so that the packet stream provides a true, albeit noisy, measure of the source clock frequency; packet sequency is unimportant; a bound on the packet jitter is known a priori (for the design of the destination buffer size); jitter correlation is not considered in the design.

We discuss the problem of clock synchronization in Section 2. We first illustrate the phenomenon of packet jitter via a simple example of a first-come first-served (FCFS) multiplexer. We develop a mathematical model of the destination buffer which describes the dynamics of its state, the number of packets in the buffer, as a function of the frequency difference between the source and destination clocks and the random packet jitter.

Section 3 describes the clock adjustment algorithms, where we propose two control algorithms - (i) a first order system, in which the current value of the destination clock frequency is obtained by adding its value to the weighted value of the frequency difference estimate at the previous instant of time; and (ii) a second order system, in which the current value of the destination clock frequency is derived from a linear combination of the values of the destination clock frequency and the frequency difference estimates at the previous two instants of time.

Section 4 contains two increasingly complex estimation algorithms for estimating the frequency difference. In the simplest suboptimal algorithm, we use no noise statistics. Therefore, this algorithm applies to all situations - packet jitter correlated or not. The next level of complexity is added when we explicitly use the signal and noise variances in the optimal Kalman filter. Since the noise variance is not known a priori, we describe a method to process the observation set of the destination buffer state to obtain the estimate of the noise variance, too. The latter algorithm is optimal with respect to an uncorrelated packet jitter.

In Section 5, we discuss the implementation aspects of these estimation and control algorithms in a broadband packet network environment. Finally, Section 6 contains some concluding remarks.

2. PROBLEM OF CLOCK SYNCHRONIZATION

Figure 1 shows a single FCFS packet multiplexer with three packet streams with rates 1.0, 2.0 and 4.0 packet per second (pps). The packets from these streams compete for the empty slots on the high speed line with a rate of 8.0 pps. In the sequel we consider the 4.0 pps rate packet stream to illustrate the packet jitter phenomenon.

Let f_1 be the constant and unknown source clock frequency in pps. Let $\phi(k)$, $k \geq 0$, denote the destination buffer level function. The reading clock frequency at the destination is $f_2(k)$ pps. $f_2(k)$ changes with k according to the adjustment laws described in the next Section. Let $\Delta f(k) = f_1 - f_2(k)$ be the difference between the source and destination clock frequencies. When the packets in the input line arrive with stochastic delays, the input packet stream represents a noisy value of the source clock frequency, and $\phi(k)$ evolves in a random manner depending on $\Delta f(k)$ and the packet jitter characteristics. The dynamics of the state of the destination buffer is obtained by the conservation law, and is given by

$$\phi(k+1) = \phi(k) + T(k)(f_1 - f_2(k)) + d(k) \qquad (2.1)$$

or

$$\Delta\phi(k) = \phi(k+1) - \phi(k) = T(k)\Delta f(k) + d(k) \qquad (2.2)$$

where $T(k) = 1/f_2(k)$ is the sampling period and $d(k)$ represents the

packet jitter at kth instant. Nonlinearity in the equation arising from the convenient choice of the sampling period is ignored. (2.1) or (2.2) is the state equation of the buffer, based on which all designs in this paper are developed.

Figures 2a, 2b, and 2c, respectively, show the plots of $\phi(k)$ as a function of k for the 4.0 pps rate packet stream when $d(k)=0$ and $\Delta f(k)=0$, when $d(k)\neq 0$ and $\Delta f(k)=0$, and when $d(k)\neq 0$ and $\Delta f(k)\neq 0$. The randomness in $\phi(k)$ due to contention at the multiplexer is obvious from these plots. (Compare Figures 2a and 2b at the sampling instants.)

For a synchronous broadband packet network, synchronized through a common master clock, f_1 is exactly known and $\Delta f(k)=0$. $\phi(k)$ in this case fluctuates from k to $k+1$ only due to the jitter term. Thus, $\Delta\phi(k)=d(k)$. (See figure 2b, for example.) $\phi(k)$ is a random variable. Let $\max_k |\phi(k)|=\delta$, $\delta>0$ for all sample functions $\phi(k)$. If the first packet from the packet stream is removed from the destination buffer after a fixed $\delta'>\delta$ (allowing for the values of $\phi(k)$ during the sampling interval) number of packets have accumulated in the buffer, then no buffer underflow will occur, and the output packet stream from the destination buffer will be uniformly spaced with a time period of $T=1/f_1$. δ is obviously related to the maximum aperiodicity introduced into the source packet stream.

For a plesiochronous broadband packet network, the source and destination clock frequencies have the same nominal value but they differ by a small amount, which is unknown and whose typical value is less than a few tenth of ppm. Some preliminary studies of multiplexer and switch design indicate that the values of $d(k)$ may vary from 0 packet to a few packets from one instant to another. Thus, the change in $\phi(k)$ is mainly due to the random jitter $d(k)$. However, since $Ed(k)=0$ ($E\cdot$ is the expectation operator.), we expect that the contribution of the $\Delta f(k)$ term to the buffer level function in $j(\cdot)T(k)$ time periods may become quite significant, where $j(\cdot)$ denotes the length of the interval, which is large and time-varying. We propose the following two time-scale model - a fast time sequence, $k\geq 0$, when we collect observation data $\phi(k)$, and a slow time sequence, $m\geq 0$, on which we compute the estimates of the frequency difference, denoted by $\Delta\hat{f}(m)$, from the set $\phi(k)$, $k\geq 0$, and adjust $f_2(m)$ using this estimated frequency difference. The slow time scale interval $[m, m+1)$ is made time-varying, i.e., $0\leq k\leq j(m)$ for the above interval. Note that there are a total of $j(m)+1$ samples of $\phi(k)$ in each large interval, and that the time length of the interval $[k, k+1]$, $T(k)=1/f_2(k)$, is constant in $[m, m+1)$ with $f_2(k)=f_2(m)$. Obviously, a better estimate of $\Delta f(k)$ would result from a larger value of $j(\cdot)$, though this would require a larger destination buffer and would cause large delays. We describe later in Section 4 how the initial interval, $j(0)$, is chosen and how $j(m)$ is varied with time.

The most difficult part in the design and analysis of the clock recovery schemes is the modeling of the noise process, $d(k)$. Its expected value, of course, is zero by the fact that if the frequency difference is zero, then the average accumulation of packets in the destination buffer must be zero, assuming no packet loss through the network. The other statistics, in general, are not known, however. $d(k)$ is expected to be a correlated sequence. The correlation is dependent on several factors - loading of the network; relative rates of the input traffic being multiplexed in comparison to the multiplexed line rate; service disciplines at the multiplexers and switches, e.g., FCFS or train scheduling; type of connection, e.g., virtual circuit or datagram. These factors will influence the statistics of the packet jitter in a complex manner. However, in general the correlation of the packet jitter will result from a repeating pattern of jitter every few packets in the stream. The period of the pattern is large if the packet stream traverses through many stages of multiplexing and switching, thereby reducing the jitter correlation. The plot of Figure 2b illustrates this repeating pattern of packet jitter every 4 packets for the example considered above.

For simplicity, in this paper we ignore the correlation of the process $d(k)$. We develop an ad hoc design (simple averaging) based on no statistics of $d(k)$, and an optimal design (Kalman filter) based on $d(k)$ being a white noise sequence.

3. CLOCK ADJUSTMENT ALGORITHMS

Since $\Delta f(m)$ is not directly available, we process the observation set $\phi(k)$, $0\leq k\leq j(m)$ in some manner to filter out the effects of the noise and obtain the frequency difference estimate, $\Delta\hat{f}(m)$, for each interval $[m, m+1)$. We use the estimate $\Delta\hat{f}(m)$ in the control algorithm to compute $f_2(m+1)$. Let $\Delta\tilde{f}(m)$ denote the frequency difference estimation error, i.e., $\Delta\tilde{f}(m) = \Delta\hat{f}(m)-\Delta f(m)$. The estimation algorithms described later are such that the estimate $\Delta\hat{f}(m)$ is unbiased, i.e., $E\Delta\hat{f}(m)=E\Delta f(m)$. Further, if $d(k)$ is a white noise sequence, then the resulting $\Delta\tilde{f}(m)$ is also a white noise sequence, since it is a linear function of $d(k)$ for all $k\in[m, m+1)$. For the purpose of analysis, we assume that. Figure 3 shows the signal flow diagram for the estimation and control algorithms, along with the two time scales.

Control Law I: The first and simple control law is to add a weighted value of the error-prone $\Delta\hat{f}(m)$ to $f_2(m)$ at m to obtain $f_2(m+1)$ at $m+1$, i.e.,

$$f_2(m+1) = f_2(m) + \alpha\Delta\hat{f}(m) \qquad (3.1)$$

for some $0<\alpha<1$. With $\Delta\hat{f}(m)$ and $\Delta f(m)$ defined as above, the closed loop system is given by

$$f_2(m+1) = (1-\alpha)f_2(m)+\alpha(f_1+\Delta\tilde{f}(m)) \qquad (3.2)$$

To compute the mean of $f_2(m)$, we take expectation on both sides of the equation (3.2). Since $Ef_1=f_1$ and $E\Delta\tilde{f}(m)=0$, this results in

$$Ef_2(m+1) = (1-\alpha)Ef_2(m) + \alpha f_1 \qquad (3.3)$$

The asymptotic value of the mean $Ef_2(\infty)$ is obtained by taking the limit as $m\to\infty$, which can be shown to yield $Ef_2(\infty) = f_1$.

The dynamics of the variance $\pi(m) = E(f_2(m)-Ef_2(m))^2$ satisfies the following equation which is derived from (3.2)

$$\pi(m+1) = (1-\alpha)^2\pi(m) + \alpha^2\sigma(m) \qquad (3.4)$$

where $\sigma(m)=E\Delta\tilde{f}^2(m)$ is the variance of the frequency difference estimation error. We have assumed that $\Delta\tilde{f}(m)$ is a white noise sequence, and thus is uncorrelated with $f_2(m)-Ef_2(m)$ since $\Delta\tilde{f}(m)$ is a function of the future values of the packet jitter, which do not influence $f_2(m)$. Since $(1-\alpha)<1$, the asymptotic variance (as $m\to\infty$) of $f_2(m)$ can be shown to be $\pi(\infty) = \frac{\alpha}{2-\alpha}\sigma(\infty)$. $\sigma(\infty)$ is the asymptotic variance of $\Delta\tilde{f}(m)$, and is related to $v(\infty)$, the asymptotic variance of the packet jitter, $d(k)$, through the estimation algorithms described later. The residual variance $\pi(\infty)$ is smaller than the asymptotic variance of the frequency difference estimation error for all $0<\alpha<1$. There is a trade-off between how fast the dynamic response should be and how much variance reduction we can obtain. A larger α means faster response ($(1-\alpha)$ is closer to zero), but higher residual variance, and vice versa. Also, a larger α implies a smaller destination buffer.

Control Law II: With the control law (3.1), there is not enough freedom to control the speed of tracking as well as the magnitude of the residual variance, $\pi(\infty)$, independently. The following control algorithm, in which the value of $f_2(m+1)$ is obtained by adding the weighted values of $f_2(m)$, $f_2(m-1)$, $\Delta\hat{f}(m)$ and $\Delta\hat{f}(m-1)$, provides more freedom in controlling the rate of convergence and the asymptotic variance. The weightings on $f_2(m)$ and $f_2(m-1)$ should be such that the open loop transfer function should contain $(1-z^{-1})$* as a factor in the denominator (so that there is no steady state error) (See [5]). With this, the new control algorithm is given by

$$f_2(m+1) = (1+\delta)f_2(m)-\delta f_2(m-1)+\alpha\Delta\hat{f}(m)+\beta\Delta\hat{f}(m-1) \qquad (3.5)$$

where δ, α and β are free parameters. The closed loop system can be obtained by inserting $\Delta\hat{f}(m)=f_1-f_2(m)+\Delta\tilde{f}(m)$ into (3.5).

We will not discuss the complete design procedure here. (See Section 5.2 of [5] for the design and pp 31 of [6] for choosing the optimum parameter values.) It can easily be shown that $Ef_2(\infty)=f_1$. The asymptotic variance, $\pi(\infty)$, can be computed from (3.5), and is given by

$$\pi(\infty) = \frac{(\alpha^2+\beta^2)(1+\beta+\delta)+2\alpha\beta(1+\delta-\alpha)}{1-(1+\delta-\alpha)^2+(\beta+\delta)(1+(\beta+\delta)+(\beta+\delta)(1+\delta-\alpha)-(\beta+\delta)^2)}\sigma(\infty)$$

By making a suitable choice of the two poles of the closed loop system to control the rate of convergence and to minimize $\pi(\infty)$, it can be shown that $\pi(\infty)$ is a monotonic and symmetric function of α and β, and the two poles. Thus, with the poles fixed at the chosen values, for $\pi(\infty)$ to be minimum, $\alpha=\beta$. The parameter α is then determined from the values of the two poles.

4. FREQUENCY ESTIMATION ALGORITHMS

The changes in $\Delta\phi(k)$ from k to $k+1$ in each interval $[m, m+1)$

* 'z' is a forward shift operator, i.e., $zx(t)=x(t+1)$.

is mainly due to the noise $d(k)$ [illegible] than the frequency difference. We exploit this separation property in obtaining the estimate of the frequency difference from the noisy observation of the rate of change of the buffer level function.

We describe below two increasingly complex algorithms for computing $\Delta\hat{f}(m)$. The first algorithm is the suboptimal simple averaging scheme and assumes nothing regarding the jitter process except that its expected value is zero. Thus, it is applicable to all systems regardless of whether $d(k)$ is correlated. The second algorithm is the Kalman filter (more accurately, Wiener filter) which assumes that $d(k)$ is a white noise sequence with zero mean and finite, but unknown, variance. Since the latter algorithm is optimal with respect to a white noise process, it may yield a poor estimate in comparison to the simple scheme in case $d(k)$ is highly correlated. While, if $d(k)$ is white, the Kalman filter will definitely provide a better estimate than the simple scheme. Both algorithms are developed assuming that control law I is used. These can easily be modified for the control law II; we do not discuss it here though.

Now, if we subtract both sides of (3.1) from f_1, then the resulting equation gives the dynamic evolution of $\Delta f(m)$

$$\Delta f(m) = (1-\alpha)\Delta f(m-1) - \alpha\Delta\hat{f}(m-1) \tag{4.1}$$

$\Delta f(m)$ responds to its initial condition $\Delta f(0)$ and the frequency difference estimation error $\Delta\tilde{f}(m-1)$. The expected value of the frequency difference satisfies

$$E\Delta f(m) = (1-\alpha)E\Delta f(m) \tag{4.2}$$

since $E\Delta\tilde{f}(m)=0$. The initial condition, $E\Delta f(0)$, is known from the prior knowledge of the frequency difference. Since $\Delta f(m)$ (and also $f_2(m)$) is uncorrelated with the future values of $d(k)$ for $k \in [m, m+1)$ of which $\Delta\tilde{f}(m)$ is a linear function, the variance of $\Delta f(m)$ is obtained from (4.1) as

$$\pi(m) = (1-\alpha)^2\pi(m-1) + \alpha^2\sigma(m-1) \tag{4.3}$$

with the initial variance $\pi(0)$ known, and $\sigma(m)$ given. This is the same as (3.4) because $\pi(m)=E(\Delta f(m)-E\Delta f(m))^2 = E(f_2(m)-Ef_2(m))^2$, since $Ef_1=f_1$. Whichever estimation algorithm we use in the interval $[m, m+1)$, the initial mean and variance of $\Delta f(m)$ for that interval are given by the above two equations (4.2) and (4.3). These quantities are explicitly used in the optimal Kalman filtering algorithm described later.

Computation of $j(\cdot)$: Now, we describe how we choose the length of the time interval $[m, m+1)$, i.e., how many data points we use for the estimation of $\Delta f(m)$ in that interval. If $j(m)$ denotes the length of the interval $[m, m+1)$, then we propose that

$$j(m+1) = \frac{1}{(1-\alpha)}j(m) \tag{4.4}$$

with $j(0)$ as the length of the initial period. It can be seen that the size of the interval increases at the same rate at which $\Delta f(m)$ decays. The rationale behind using this scheme is that as the frequency difference decreases with increasing time, the signal to noise ratio decreases. Hence, we should have an increasingly larger interval for a better estimate of the decreasing $\Delta f(m)$ with m.

Both algorithms provide an estimate of $\Delta f(m)$ by processing the finite data set $\Delta\phi(k)$, $0 \le k \le j(m)$ in the interval $[m, m+1)$. For convenience, the buffer state equation (2.2) is rewritten here as

$$\Delta\phi(k) = \frac{\Delta f(m)}{f_2(m)} + d(k) \tag{4.5}$$

Estimation Algorithm I: An ad hoc scheme to estimating $\Delta f(m)$ may be just to take the sample mean of all the available observations $\Delta\phi(\cdot)f_2(m)$. This is motivated by the idea that the zero mean noise samples will cancel out by averaging it. There are a total of $j(m)$ samples in the interval $[m, m+1)$, and thus,

$$\Delta\hat{f}(m) = \frac{1}{j(m)}\sum_{k=0}^{j(m)-1}\Delta\phi(k)f_2(m) \tag{4.6}$$

which after simplification gives

$$\Delta\hat{f}(m) = \frac{f_2(m)}{j(m)}(\phi^-(m+1)-\phi(m)) \tag{4.7}$$

where $\phi^-(m+1)$ is the value of $\phi(\cdot)$ just before the control action is taken at the instant $m+1$. For this algorithm, there is no quantitative way to determine how good the estimate is at the end of each estimation period, since the noise statistics is not known. However, it is possible [illegible] draw some conclusions regarding the asymptotic behavior of the estimator, if $d(k)$ is assumed to be white.

Consider the error sequence, $\Delta\tilde{f}(m) = \frac{f_2(m)}{j(m)}\sum_{k=0}^{j(m)-1} d(k)$, for all $m \ge 0$. $E\Delta\tilde{f}(m)=0$ since $f_2(m)$ is independent of $d(k)$ for all k in $[m, m+1)$ and $Ed(k)=0$. By the same reasoning, the variance of estimation error, $\sigma(m)=E\Delta\tilde{f}^2(m)$, is given by $\sigma(m) = \frac{Ef_2^2(m)}{j^2(m)}\sum_{k=0}^{j(m)-1} v(k)$, where $v(k)=Ed^2(k)$ is the variance of $d(k)$. If the noise variance is bounded, i.e., $v(l) \le \bar{v}$, for all $l \ge 0$ and for some finite positive constant $\bar{v}$, the estimation error variance is, then, bounded by $\sigma(m) \le \frac{Ef_2^2(m)\bar{v}}{j(m)}$. Now, as $j(m)\to\infty$ with $m\to\infty$, $\sigma(m)\to 0$, since $Ef_2(m)$ is a finite quantity. Thus, the frequency difference estimation error variance asymptotically goes to zero. Asymptotic results are good for analysis. In practice, $j(m)$ is settled at some constant value for $m \ge m_1$, where m_1 is fixed on the basis of some criterion discussed in Section 5. $\sigma(\infty)$ in this case is a finite but small quantity.

Estimation Algorithm II: Further reduction in the variance of the estimation error sequence, $\sigma(m)$, $m \ge 0$ is possible by making use of the variances of $\Delta f(m)$ and $d(k)$. The variance, $\pi(m)$, of $\Delta f(m)$ for each m is obtained by the recursive dynamic equation (4.3), since $\pi(0)$ is assumed to be known and $\sigma(m-1)$ is known from the previous estimation interval $[m-1, m)$. The noise is assumed to be white, with unknown variance. We assume that the noise remains stationary in each interval $[m, m+1)$, although, as discussed in the Appendix, it is not necessary. The Appendix contains a mechanism by which the variance of the noise is estimated in real time using the same set of data $\Delta\phi(\cdot)$.

Quasi-stationary Noise Process: Let $v(m)$ be the variance of $d(k)$ in $[m, m+1)$. Let $\hat{v}(m)$ be its estimate. The algorithm for computing $\Delta\hat{f}(m)$ is derived in the Appendix, and is given by

$$\Delta\hat{f}(m) = \frac{p(m)}{1+j(m)p(m)}(\phi^-(m+1)-\phi(m))f_2(m) + \frac{1}{1+j(m)p(m)}E\Delta f(m)$$

where $p(m) = \frac{\pi(m)}{Ef_2^2(m)\hat{v}(m)}$. $\phi^-(m+1)$ is as defined above. $E\Delta f(m)$ is known from (4.2), which is derived from the known quantity, $E\Delta f(0)$. The variance of the error sequence, $\sigma(m)$, is shown to be

$$\sigma(m) = \frac{\pi(m)}{1+j(m)p(m)} \tag{4.8}$$

where $\pi(m)$ is obtained as explained above via (4.3). $\hat{v}(m)$ is obtained via the methods described in the Appendix (Equation (A.4) or (A.5)). This value of $\sigma(m)$ is used in (4.3) to generate $\pi(m+1)$. The value of $Ef_2(m)$ can be approximated with $f_2(m)$ (or with some known quantity) (see Section 5 below) in the above algorithm.

Approximation with Stationary Noise Process: If the noise variance is assumed to be constant, say, $\bar{v}$, for all $m \ge 0$, then the algorithm can be simplified as follows. Further, approximate $Ef_2(m)=f_2(0)$ (see Section 5 below), since $\Delta f(0)$ is a small quantity. Now, $p(m) = \frac{\pi(m)}{f_2^2(0)\bar{v}}$. In this case, a dynamic equation for $p(m)$ is obtained from its definition and the equations (4.3) for $\pi(m)$ and (4.8) for $\sigma(m)$.

$$p(m+1) = (1-\alpha)^2 p(m) + \alpha^2\frac{p(m)}{1+j(m)p(m)} \tag{4.9}$$

with $p(0) = \frac{\pi(0)}{f_2^2(0)\bar{v}}$ known. Since $\bar{v}$ is not known, we propose to use the variance estimation algorithm, (A.4) or (A.5), to estimate the constant variance, $\bar{v}$, in the interval $[0,1)$, which will be used during the other intervals as well. The equation (4.9) is shown to be stable for all bounded $\hat{v}(m)$ and $Ef_2(m)$ in the Appendix.

It can now be seen that for all bounded $p(m)$ and $\hat{v}(m)$, $\sigma(m)\to 0$ as $j(m)\to\infty$ with $m\to\infty$. Hence, this algorithm, too, has the same desired asymptotic behavior as the previous algorithm. It should be emphasized that if the sequence $d(k)$ is highly correlated, this algorithm may give poor performance.

5. IMPLEMENTATION OF ALGORITHMS

The clock frequency is usually expressed as its average value in bits per second, along with its tolerance in standard deviation in ppm. We will use the packets per second unit for both quantities assuming a fixed length packet. Let the source clock frequency be

$f_1 = f \pm \epsilon_t$, where f is the nomin[illegible]eqnency and ϵ_t is its standard deviation. Likewise, let the initial destination clock frequency be $f_2(0) = f \pm \epsilon_d$. This suggests that $Ef_1 = Ef_2(0) = f$, and the variance of f_1 is ϵ_t^2 and that of $f_2(0)$ is ϵ_d^2. Therefore, $E\Delta f(0) = 0$ and $\pi(0) = \epsilon_t^2 + \epsilon_d^2$.

Now, the next question is how to choose $j(0)$? $j(0)$ is determined from the prior knowledge of $\Delta f(0)$ (or $\pi(0)$), the maximum value of the noise magnitude and the allowable buffer size, so that even if the system is running for $j(0)$ units of time in the presence of noise, there is no buffer under- or overflow, and we get a reasonable estimate of $\Delta f(0)$ at the end of the first interval. In practice, after a few iterations $\Delta f(m)$ will become very small. At this time we can stop changing $j(m)$ and use a constant value. We propose that for some $m = m_1$, whenever either $|\Delta \hat{f}(m_1)| < \epsilon_1$, or $\pi(m_1) < \epsilon_2$, or $\frac{\pi(m_1)}{v(m_1)} < \epsilon_3$, then set $j(m) = j(m+1)$ for all $m \geq m_1$, where ϵ_1, ϵ_2 and ϵ_3 are design parameters.

Resetting of Algorithms: Resetting may be necessitated whenever clocks at a node are switched or replaced. We propose the following mechanism. Any time the value of the sample function $|\phi(k)| \geq \rho$, we reset the algorithm with the same set of initial conditions. The assumption is that the clocks have the same kind of freqnency tolerance. $\pm\rho$ is the upper and lower limits on the buffer size, and can be determined on the basis of the maximum expected deviations in $\phi(k)$ due to the packet jitter, the expected values of $j(m_1)$, $\Delta f(m_1)$, $f_2(m_1)$, and the changed freqnency difference. As long as we can choose all the parameters conservatively to avoid buffer over- or underflow, a false resetting of the algorithms makes little difference in the steady state.

In the above algorithms, the time intervals $[m, m+1)$, $m \geq 0$ are determined a priori, and the destination buffer levels are read at time instants, $m \geq 0$, the values of which may contain a fraction of packets. Alternatively, it is also possible to fix the destination buffer level function at discrete packet levels, e.g., half, three quarter etc., and to measure the time intervals for $\phi(k)$ to attain these predetermined labeled values. There are two limitations of this technique: (I) $\phi(k)$ may occasionally cross the labeled points due to a large bnt unexpected jitter. If for some reason these crossings occur too frequently, the estimation of the freqnency difference may be quite poor due to small number of observations, which will result in a larger residnal jitter. (II) By observing $\phi(k)$ at a coarse level, we loose information abont the statistics of the high frequency jitter component buried in it. Thus, the variance of the jitter cannot be estimated on-line from this observation set, and the Kalman algorithm cannot be used.

If $d(k)$ is highly correlated, then the control law II with the estimation algorithm I would be a preferred combination. This is because the residual effects on $f_2(m)$ due to the poor estimates $\Delta \hat{f}(m)$ can be further filtered ont better by the second control law than the first one.

6. CONCLUDING REMARKS

We have developed clock recovery schemes for the real-time traffic in a broadband packet network. The algorithms for the adjustment of the clock freqnency are based on a mathematical model of the destination buffer, and the assumptions made regarding the nature of the packet jitter introdnced by the network on a periodic packet stream generated by the source. We process the same observation set, the destination buffer level function, to obtain the estimates of beth quantities - the frequency difference between the source and destination clocks, and the jitter statistics; we perform some sort of averaging to obtain the former, while a sort of differencing yields the estimate of the jitter variance. It has been analytically shown that the destination clock freqnency adjnsted according to the proposed algorithms follows the source clock frequency asymptotically, with a minimum of fluctuation.

We have provided a systematic design procedure for choosing the relevant parameters for implementation, and have also discussed the sensitivity of the performance with respect to these parameters. We plan to test the validity and effectiveness of the algorithms via simulation by implementing them on a compnter model of a single stage FCFS multiplexer for different source traffic rates, which we will report in a future paper.

REFERENCES

[1] A. Thomas, J. P. Coudrense and M. Servel, "Asynchronous Time-Division Techniques: An Experimental Packet Network Integrating Video Communication," International Switching Symposium, Florence, Italy, May 1984.

[2] L. T. Wu, S. H. Lee and T. T. Lee, "Dynamic TDM - A Packet Approach to Broadband Networking," IEEE International Conference on Communications, Seattle, WA, June 1987.

[3] J. Y. Cochennec, P. Adam and T. Houdoin, "Asynchronous Time Division Networks: Terminal Synchronization for video and sound signals," IEEE GLOBECOM'85, New Orleans, LA, November 1985.

[4] M. De Prycker, et al, "Terminal Synchronization in Asynchronous Networks, IEEE International Communications Conference, Seattle, WA, June 1987.

[5] G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*, Addison-Wesley Publishing Company, 1980.

[6] F. M. Gardner, *Phaselock Techniques*, John Wiley & Sons, New York, 1979.

[7] I. B. Rhodes, "A Tutorial Introduction to Estimation and Filtering," *IEEE Transactions on Automatic Control*, Vol. AC-16, No. 6, Dec. 1971, pp 688-706.

APPENDIX

Kalman Filter: The Kalman filter is the best example of an optimal filtering algorithm based on the mean square criterion and a white noise sequence. The reference [7] provides a good tutorial introduction to the Kalman filter.

$\Delta f(m)$ and $f_2(m)$ are constant in the interval $[m, m+1)$. Thns, for the development of the Kalman filter algorithm the state and observation equations in that interval are given by

$$\Delta f(k+1) = \Delta f(k)$$

$$\Delta\phi(k) f_2(m) = \Delta f(k) + f_2(m) d(k)$$

The Kalman filter eqnation with the estimation error, $\Delta \tilde{f}(k)$, $(\Delta \tilde{f}(k) = \Delta \hat{f}(k) - \Delta f(k))$ is

$$\Delta \hat{f}(k+1) = \Delta \hat{f}(k) + g(k+1)(\Delta\phi(k+1) f_2(m) - \Delta \hat{f}(k)) \qquad (A.1)$$

with

$$\Delta \hat{f}(0) = E\Delta f(m) + g(0)(\Delta\phi(0) f_2(m) - E\Delta f(m)) \qquad (A.2)$$

as the initial condition, where $E\Delta f(m) = (1-\alpha)^m E\Delta f(0)$ is the expected value of the initial freqnency difference, which is known. The gain, $g(k)$, and the estimation error variance, $\sigma(k) = E\Delta \tilde{f}^2(k)$, are given by

$$g(k) = \frac{\sigma(k)}{Ef_2^2(m)v(k) + \sigma(k)} \quad \text{and} \quad \sigma(k+1) = \frac{Ef_2^2(m)v(k)\sigma(k)}{Ef_2^2(m)v(k) + \sigma(k)} \qquad (A.3)$$

with $\sigma(0) = E\Delta \tilde{f}^2(0) = E(\Delta f(m) - E\Delta f(m))^2 = \pi(m)$ as the initial condition, which is also available from the recursive equation (4.3). $Ef_2^2(m)v(k)$ is the variance of $f_2(m)d(k)$, since $f_2(m)$ is independent of $d(k)$ for all $k \in [m, m+1)$, and $Ed(k) = 0$.

If the noise is nonstationary, then (A.1), (A.2) and (A.3) describe the recursive Kalman filter algorithm with $v(k)$ replaced with its estimate, $\hat{v}(k)$, obtained through the methods described later. For simplicity, we assnme that the noise is piecewise nonstationary, i.e., its variance is constant in each interval $[m, m+1)$. In this case, the filter equation can be solved analytically to obtain a simple nonrecursive estimator to compute $\Delta \hat{f}(m)$.

The analytic solution of the above difference equation for $\sigma(k)$ is

$$\sigma(k) = \frac{\sigma(0)}{1 + \frac{\sigma(0)}{Ef_2^2(m)\hat{v}(m)} k}$$

with $\sigma(0) = \pi(m)$. $\hat{v}(m)$ denotes the estimate of the constant noise variance in the interval $[m, m+1)$. Now, with this $\sigma(k)$ and $g(k)$ from above, the filter equation (A.1) can be solved to yield

$$\Delta \hat{f}(m) = \frac{p(m)}{1 + j(m)p(m)}(\phi^-(m+1) - \phi(m)) f_2(m) + \frac{1}{1 + j(m)p(m)} E\Delta f(m$$

with $p(m)=\dfrac{\pi(m)}{Ef_2^2(m)\hat{v}(m)}$. $\phi^-(m+1)$ is the value of the buffer level function at the instant just before $m+1$, and $\pi(m)$, $\hat{v}(m)$, $E\Delta f(m)$, $j(m)$ and $f_2(m)$ are all known quantities. For the purpose of computation, we will take $Ef_2(m)=f_2(m)$.

On-line Computation of Noise Variance Estimate, $\hat{v}(k)$*:* Now, we develop a method for estimating $v(k)$ from the available information, $\Delta\phi(k+1)$, $\Delta\phi(k)$, ..., $\Delta\phi(0)$.

Approximate Analysis: Since $\Delta f(k)$ is small, we can ignore its effect on $\Delta\phi(k)$ on the fast time scale. Therefore, $\Delta\phi(k)=d(k)$ for all $k\geq 0$. Hence, $v(k)=Ed^2(k)=E\Delta\phi^2(k)$. Since the probability density function of $\Delta\phi(k)$ is unknown, the variance, $v(k)$, can be estimated as follows ($\hat{v}(k)$ is the estimate of $v(k)$ at k.)

$$\hat{v}(k+1)=\frac{k}{k+1}\hat{v}(k)+\frac{1}{k+1}\Delta\phi^2(k) \qquad (A.4)$$

for all $\infty>k\geq 0$, with $\hat{v}(0)=\Delta\phi^2(0)$, or some initial guess of $\hat{v}(0)$, if it is available.

Exact Analysis: Since $f_2(m)$ and, therefore, $\Delta f(m)$ are both constant in the interval $[m, m+1)$, successive differencing of the observation $\Delta\phi(\cdot)$ yields

$$\Delta\Delta\phi(k)=\Delta\phi(k+1)-\Delta\phi(k)=d(k+1)-d(k)$$

The expected value of $\Delta\Delta\phi(k)$ is zero, and its variance is given by $E\Delta\Delta\phi^2(k)=E(d(k+1)-d(k))^2$. Since the noise is uncorrelated, this yields

$$v(k+1)+v(k)=Ed^2(k+1)+Ed^2(k)=E\Delta\Delta\phi^2(k)$$

$v(k)$ in general is time-varying, but if we assume that its value does not change significantly from k to $k+1$, then $v(k+1)\approx v(k)$. Thus,

$$v(k)=Ed^2(k)=\frac{1}{2}E\Delta\Delta\phi^2(k)$$

As before, $\hat{v}(k)$ can be computed from the following recursive equation

$$\hat{v}(k+1)=\frac{k}{k+1}\hat{v}(k)+\frac{1}{k+1}\,\frac{1}{2}\Delta\Delta\phi^2(k) \qquad (A.5)$$

with $\hat{v}(0)=\dfrac{\Delta\Delta\phi^2(0)}{2}$ as the initial condition. The recursive computation starts at $m=0$, and must be frozen at time instants $j(m)$ and $j(m)-1$ because in the definition of $\Delta\Delta\phi(\cdot)$ two future values of $\phi(\cdot)$ are needed and the value of $\Delta f(m)$ changes at the time instant m due to the control action. Here we assume that the effect of this control action on $\phi(\cdot)$ is observed only at the next time instant, i.e., one fast time-scale time period later.

If the noise statistics is such that the variance $v(\cdot)$ is a constant, then the recursive variance estimator will come to a steady state value quite soon, perhaps in the first slow time scale interval. At this time, the estimation of the noise variance can stop. Which variance estimation algorithm should be used will depend on the relative magnitude of $\Delta f(0)$ and the expected packet jitter.

Problem of Stability: There is a potential problem of stability due to the time-varying nature of the variance estimator, and $\sigma(m)$ being generated by the nonlinear equation (A.3) and used in the computation of $\pi(m+1)$ via (4.3). $\hat{v}(m)$ computed via (A.4) or (A.5) is a bounded quantity, i.e., $0<\hat{v}(m)\leq\bar{v}$, for some constant $\bar{v}$. $Ef_2(m)$ is also a bounded quantity, i.e., $|Ef_2^2(m)|\leq\bar{f}_2$, for some finite constant $\bar{f}_2$. Then it is easy to see that, with $p(m)=\dfrac{\pi(m)}{\bar{f}_2\bar{v}}$, the equation for $p(m)$ is obtained from (4.3) as

$$p(m+1)\leq\left((1-\alpha)^2+\frac{\alpha^2}{1+j(m)p(m)}\right)p(m)$$

It is not difficult to show that if $0<\alpha<1$, then the coefficient of $p(m)$ in the above equation is less than one for all $p(m)$, bounded or unbounded. Hence, $p(m)\to 0$ as $j(m)\to\infty$ with $m\to\infty$.

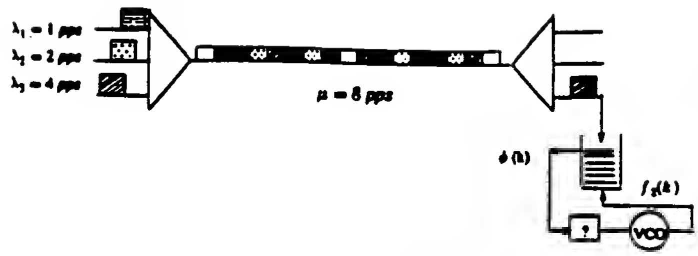

Figure 1 : A Network with a FCFS Multiplexer

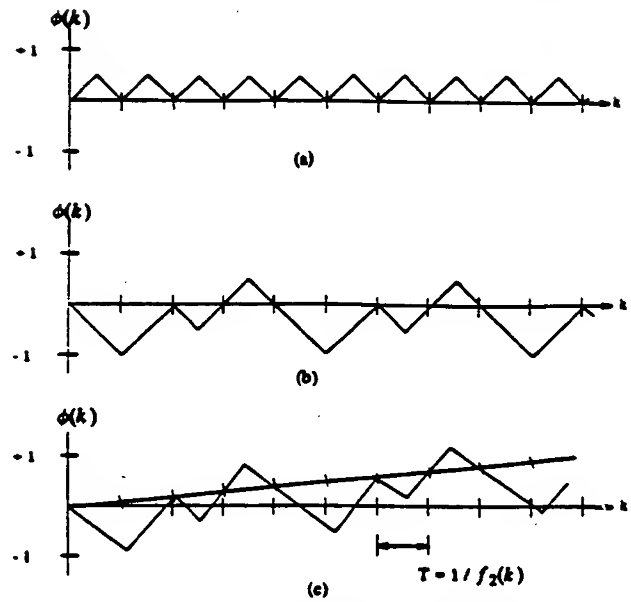

Figure 2 : Buffer Filling Level as a Function of Time

(a) No contention and $\Delta f = 0$

(b) Contention and $\Delta f = 0$

(c) Contention and $\Delta f \neq 0$

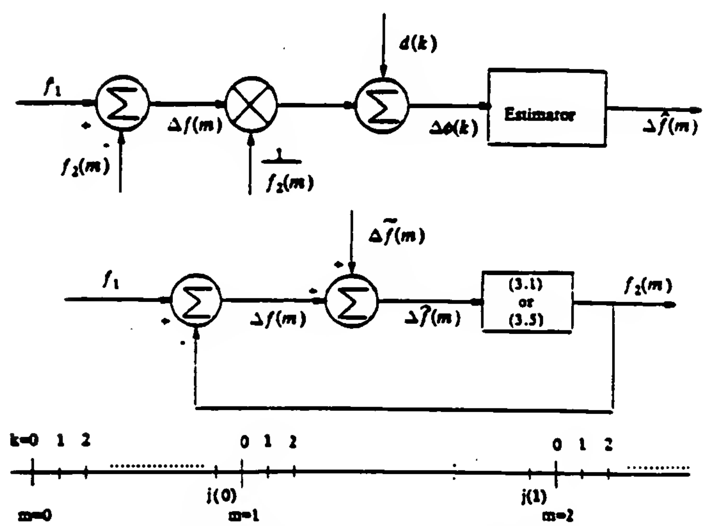

Figure 3 : Signal Flow Diagram for Estimation and Control

Algorithms on a Two Time Scale